

UNI-JUMP

Diagramme des classes UNI-JUMP

Ely Wandeu | May 1, 2023

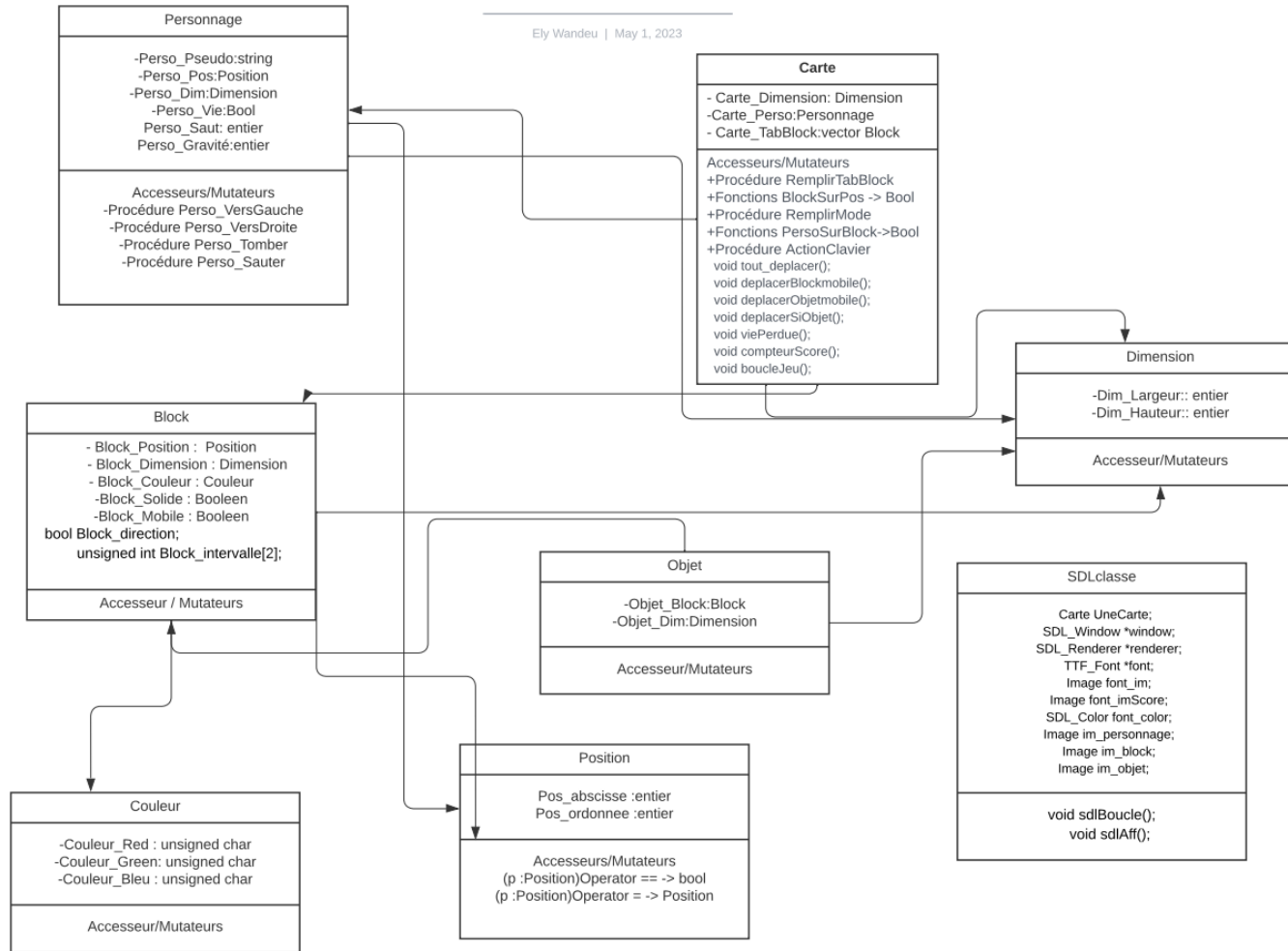


Diagramme des classes

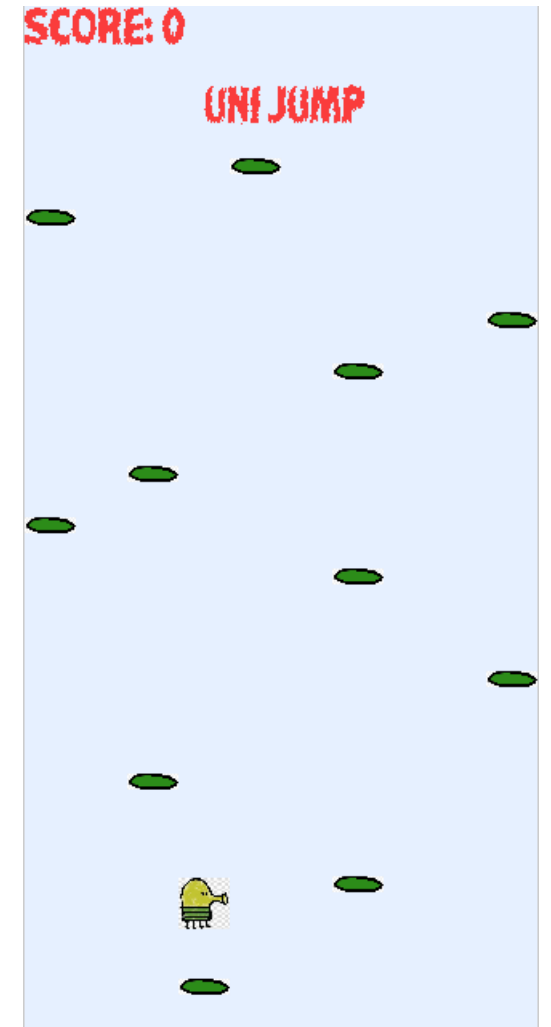
UNI-JUMP:

Crée par:

- WANDEU Ely
- HAFFOUDHI Mohamed samir
- OUKRAOU Aissam

C'est quoi:

C'est un jeu dans lequel le joueur cherche à monter le plus haut possible en s'appuyant sur des barres qu'on appelle des **Blocks**, qui peuvent être des Blocks **bougeants** ou **figés**. à travers le jeu, le joueur fera face à toutes sortes d'obstacles et dangers qu'il doit éviter afin de ne pas tomber



Classe personnage

La classe personnage c'est la classe où il contient le information de joueur , son nom, sa dimension, sa vie et le saut maximum qu'il peut faire.

```
// nom du personnage
std::string Perso_pseudo;
// position du personnage
Position Perso_Position;
// dimension du personnage
Dimension Perso_Dim;
// nom fichier image du personnage
// std::string Perso_image;
// si est vivant ou mort
bool Perso_Vie;

// nombre de pixel par carreau
unsigned int nbPixelParCarreau;
// puissance de saut, hauteur de chaque saut. Nb de carre monté par saut
unsigned int Perso_Saute;
```

Classe Block

La classe block est une classe particulier, puisque est utilisé deux fois , une fois pour créer les étagères et une fois pour créer les objets (cette à dire le trampoline).

Elle contient la position, la dimension, le couleur, si elle est solide (que pour les étagères) , si elle est mobile et si oui son intervalle.

```
private:
```

```
    Position Block_Position;
```

```
    Dimension Block_Dimension;
```

```
    Couleur Block_Couleur;
```

```
    //Si c'est égal a true on peut se poser dessus. Sinon on le casse et on tombe
```

```
    //si Block_direction est égal a true alors le block va vers la gauche sinon vers la droite
```

```
    bool Block_Solide;
```

```
    bool Block_mobile;
```

```
    bool Block_direction;
```

```
    unsigned int Block_intervalle[2];
```

Classe carte:

La classe carte c'est la plus important où est réalisée le jeu et il commande toutes les autres classes, elle contient la dimension de la carte, le personnage, un tableau de vecteur de type Block pour les étagères, un tableau de vecteur de type Block pour les objets, le nombre saut de personnage, le score et un entière objet pour quand le personnage touche des objets.

```
private:
    //dimension de la carte
    Dimension Carte_Dimension;

    //Personnage dans la carte
    Personnage Carte_perso;

    //Tableau dynamique de block
    std::vector<Block> Carte_TabBlock;

    //Tableau dynamique d'objet
    std::vector<Block> Carte_TabObjet;

    // int Carte_taille_TabBlock;
    unsigned int nbSautmax;

    //Score de jeu quand il tape un block
    unsigned int score;

    //bool objet
    unsigned int objet;
```

Conclusion

Objectif du projet était de simuler le jeu pour téléphone comme doodle jump. On avait pensé de faire beaucoup de fonctionnalité qui dépendait du temps et du parcours, mais que au finale était un peu plus difficile de prévu, on a perdue du temps plutôt sûr comment le réaliser, recherche les erreurs et l'organisation des tâches. Mais on peut dire que au final on beaucoup appris de cette expérience. Maintenant on sait un peu mieux programmer, à s'organiser sûr les différentes tâches. Pour la prochaine fois on sait déjà que il faudrait mieux s'investir.